Corrigendum: Sampling regular graphs and a peer-to-peer network

Colin Cooper
Department of Computer Science
Kings College
London WC2R 2LS, U.K.
ccooper@dcs.kcl.ac.uk

Martin Dyer
School of Computing
University of Leeds
Leeds LS2 9JT, U.K.
dyer@comp.leeds.ac.uk

Catherine Greenhill
School of Mathematics and Statistics
The University of New South Wales
Sydney NSW 2052, Australia
csg@unsw.edu.au

21 February 2012

Abstract

In [*Combinatorics, Probability and Computing* **16** (2007), 557–593, Theorem 1] we proved a polynomial-time bound on the mixing rate of the switch chain for sampling d-regular graphs. This corrigendum corrects a technical error in the proof. In order to fix the error, we must multiply the bound on the mixing time by a factor of d^8.

For notation and terminology not defined here, see [1]. Let $\Omega_{n,d}$ be the set of all d-regular graphs on the vertex set $\{1, 2, \ldots, n\}$. A natural Markov chain for sampling elements of $\Omega_{n,d}$ is studied in [1]. We will refer to this chain as the *switch chain*. The transition procedure is given in [1, Figure 1]: from a current state in $\Omega_{n,d}$ it chooses two non-incident edges uniformly at random, and with probability $\frac{1}{2}$ it chooses a perfect matching of the four endvertices uniformly at random, replacing the two chosen edges with this perfect matching unless multiple edges would result. (With the remaining probability $\frac{1}{2}$ it does nothing, as this is a lazy chain.) Our main theorem stated that the mixing time of this chain is bounded above by

$$\tau(\varepsilon) \le d^{15}\, n^8\, (dn \log(dn) + \log(\varepsilon^{-1})). \qquad (1)$$

There was a slight error in the proof of one of the lemmas used to establish this result, namely [1, Lemma 5]. Correcting the lemma leads to the new upper bound given below, which is a factor of d^8 larger. (It is possible that with a more sophisticated argument a smaller power of d might be enough. But we choose to use a simple argument here.)

Theorem 1. *Let $\tau(\varepsilon)$ be the mixing time of the switch Markov chain with state space $\Omega_{n,d}$. Then*

$$\tau(\varepsilon) \leq d^{23}\, n^8\, (dn \log(dn) + \log(\varepsilon^{-1})).$$

The error arose in [1, Lemma 5], which bounds the flow $f(e)$ carried by an arbitrary transition e of the switch Markov chain. We give a corrected statement of this lemma below and explain the changes required to correct the proof.

Lemma 1. *For any transition $e = (Z, Z')$ of the Markov chain,*

$$f(e) \leq 2d^{20} n^5 / |\Omega_{n,d}|.$$

Proof. Fix a transition $e = (Z, Z')$ of the switch chain, and let $(G, G') \in \Omega_{n,d} \times \Omega_{n,d}$ be such that e lies on the canonical path $\gamma_\psi(G, G')$ from G to G' corresponding to the pairing $\psi \in \Psi(G, G')$. From e and (G, G', ψ), the encoding L and a yellow-green colouring of $H = G \triangle G' = Z \triangle L$ can be formed, with green edges belonging to Z and yellow edges belong to L. As in the proof of [1, Lemma 5], the pairing ψ produces a circuit decomposition $\mathcal{C}$ of H with colours alternating yellow, green almost everywhere.

Call a vertex x *bad* with respect to ψ if two edges of the same colour are paired at x under ψ. If a vertex is not bad, it is called *good*. Every bad vertex lies on the circuit currently being processed. Specifically, bad vertices may only be found next to *interesting edges*, which are

- odd chords which have been switched during the processing of a 1-circuit (but not switched back), or
- the shortcut edge, switched while processing a 2-circuit (but not yet switched back).

Recall that an edge is *bad* if it receives label -1 or 2 in the encoding L. Note that all bad edges are interesting (an interesting edge is bad if and only if it does not lie in the symmetric difference H). Therefore the interesting edges form a subgraph of one of the graphs shown in [1, Figure 10], and [1, Lemma 2] also describes interesting edges: there are at most four interesting edges and hence at most six bad vertices.

A yellow-yellow or green-green pair at a bad vertex x is called a *bad pair*. In the proof of [1, Lemma 5] we proved that two quantities $|\Psi'(L)|$ and $|\Psi(H)|$ are related by the inequality

$$|\Psi'(L)| \leq d^b\, |\Psi(H)| \tag{2}$$

where b denotes the maximum number of bad pairs in any encoding L. To prove this we showed that each bad pair contributes a factor of at most d to the right hand side of this inequality. In [1] this equation appeared as [1, Equation (6)] with $b = 6$, as we claimed that there are at most 6 bad pairs in any encoding. As we will see, the

true maximum value is $d = 14$. We explain this below, and then extend the argument from [1] to prove that each bad pair contributes a factor of at most d to the right hand side of (2). Hence [1, Equation (6)] can be replaced by

$$|\Psi'(L)| \leq d^{14}\, |\Psi(H)|, \tag{3}$$

giving an extra factor of d^8.

First we explain why there can be up to 14 bad pairs in an encoding. It is true that, in the circuit C currently being processed, a bad pair will occur at each bad vertex which is the endvertex of an interesting edge. This accounts for up to six bad pairs. However, in [1] we overlooked the fact that the interesting edges may themselves belong to the symmetric difference H. For each such interesting edge there will be a bad pair at each endvertex of the edge, in the circuit containing the edge. As there are at most four interesting edges, this gives *up to eight additional bad pairs*, giving a total of at most 14 bad pairs in any encoding. (An example is given below which shows that the upper bound of 14 can be achieved.)

Next we explain why (2) still holds, now with $b = 14$. In the proof of [1, Equation (1)] we stated that a bad vertex was incident with at most one bad pair of each colour. This limited the cases that we considered when calculating the number of ways to pair up the edges around each bad vertex. Now when the eight additional bad pairs are taken into consideration, we see that it is possible to have up to two bad pairs of each colour present at a bad vertex. But these extra cases can be dealt with using very similar arguments to those in [1], showing that every bad pair contributes a factor of at most d to the number of pairings at that vertex. For example, suppose that a vertex v is incident with $\theta_v + 2$ green arcs and $\theta_v - 2$ yellow arcs. Then v is bad, with two bad green pairs and no bad yellow pairs. The number of ways to pair up the edges around v is then

$$3\binom{\theta_v + 2}{4}(\theta_v - 2)! = \frac{(\theta_v + 2)(\theta_v + 1)}{8}\, \theta_v! \leq \theta_v^2\, \theta_v! \leq d^2\, \theta_v!.$$

This extra factor of d^2 arises from the two bad pairs at v, as required. All other cases can be dealt with similarly. This shows that (3) holds. The extra factor of d^8 carries through the rest of the proof of [1, Lemma 5], leading to the corrected bound stated here. □

Proof of Theorem 1. The additional factor of d^8 from Lemma 1 is carried throughout the rest of the proof of Theorem 1. For example, the bound on the load of the flow, $\rho(f)$, given in [1, Equation (7)] must become

$$\rho(f) \leq 2d^{22}\, n^7.$$

Finally, the extra factor d^8 appears in the bound on the mixing time, giving us (1). □

We now work through an example which shows that there can indeed be 14 bad pairs in the yellow-green colouring of H. (See [1] for a detailed description of the procedures used.)

The symmetric difference H that we use is shown in Figure 1. Initially, solid lines belong to G and dashed lines belong to G'. One additional edge, x_0x_5, is shown using a dotted arc: this edge belongs to $G \cap G'$ but will be used to create the canonical path, so we include it in our figures.

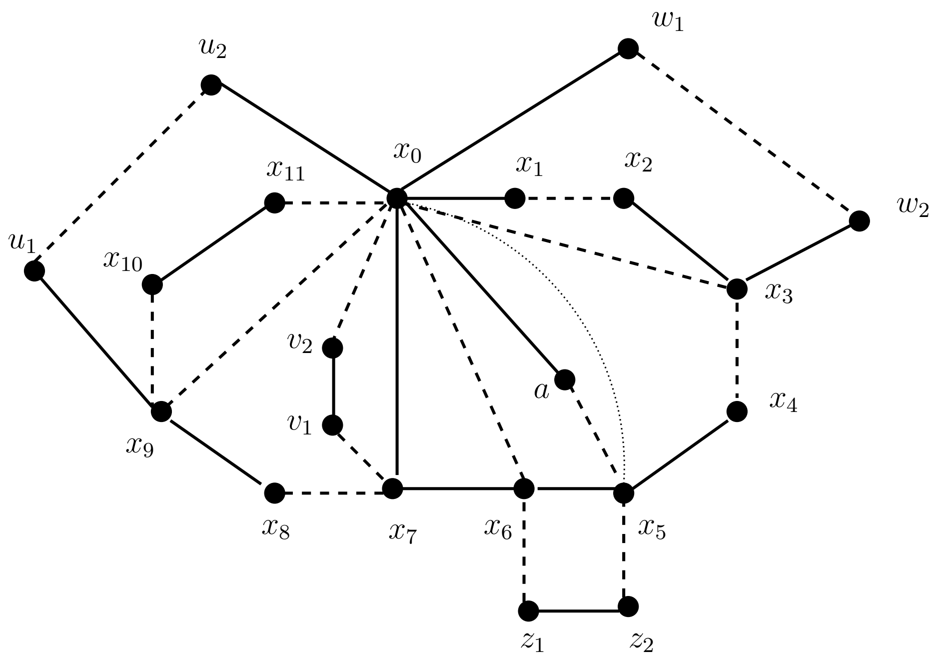

Figure 1: The symmetric difference H (solid for Z and dashed for $H - Z$) together with one additional edge (dotted). Initially $Z = G$ and $H - Z = G'$.

Suppose that we work with the pairing ψ that produces the following sequence of circuits $\mathcal{C}$:

$$x_0ax_5x_4x_3x_2x_1x_0x_{11}x_{10}x_9x_8x_7x_6, \ \ x_0x_9u_1u_2, \ \ x_0w_1w_2x_3, \ \ x_0x_7v_1v_2, \ \ x_5x_6z_1z_2.$$

These circuits are processed in the given order, so the first circuit to be processed is the longest one. It is a 2-circuit in case (b) with shortcut edge x_5x_6. (To match up the notation with [1, Figure 3], use the transformation $(x_0, a, x_5, x_6) \mapsto (v, w, x, y)$: our choice of notation will prove convenient when processing the 1-circuit arising from this 2-circuit.) After performing the shortcut switch we obtain the situation of Figure 2. Interesting edges will be shown very thickly, and bad vertices will also be drawn larger. Bad pairs are indicated by thin arcs joining the two edges in the pair. For example, vertices x_6 and x_5 are bad vertices in Figure 2 with one bad pair of each colour at each vertex. The shortcut edge $\{x_6, x_5\}$ is interesting.

Next we must process the 1-circuit containing the shortcut edge, which is

$$x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}.$$

There will be three phases, and the first phase only takes one step, producing Figure 3. We see two interesting edges (the shortcut edge and the odd chord), four bad vertices

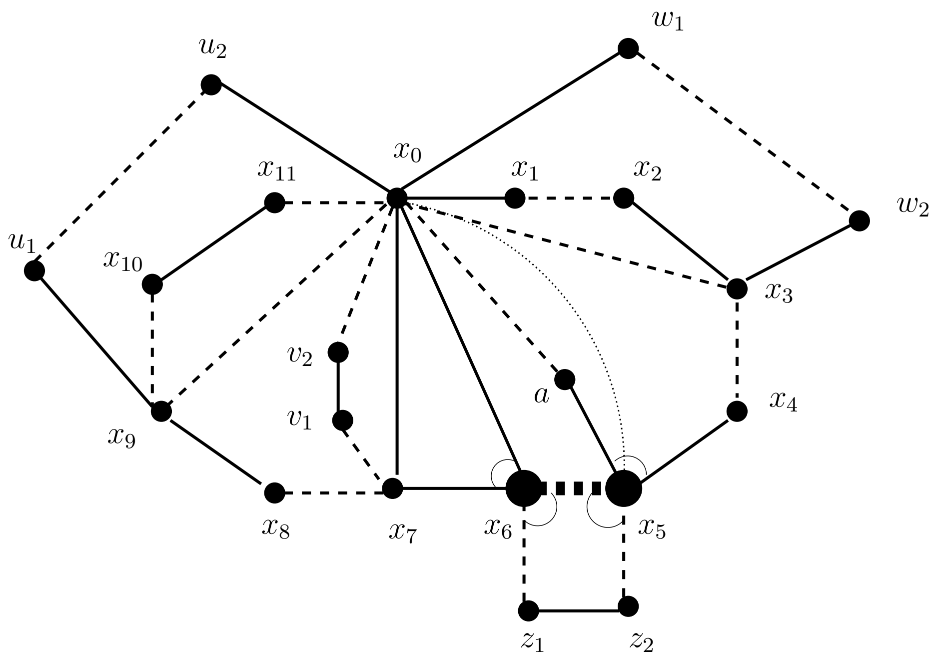

Figure 2: After the shortcut switch, with 1 interesting edge, 2 bad vertices, 4 bad pairs.

and eight bad pairs.

Next we launch into Phase 2, and after one step we see the situation of Figure 4. Now there are three odd chords as well as the shortcut edge, giving four interesting edges. None of them are bad as we assumed that they belong to H. But this means that once they are disturbed, they each contribute two bad pairs. Overall the six bad vertices contribute 14 bad pairs. At x_0 we see two bad pairs of each colour, and elsewhere we see one bad pair of each colour.

Take one more step of Phase 2 to obtain Figure 5. The shortcut edge is now restored, so we just have the three odd chords. Two of these belonged to the symmetric difference originally and hence contribute two bad pairs each (one at each endpoint). The other, x_0x_5 was assumed to belong to $G \cap G'$ and does not contribute any bad pairs. (If it also belonged to the symmetric difference then we would have 10 bad pairs at this step.)

Now we complete Phase 2 in one more step, see Figure 6.

We only have one interesting odd chord left, giving two bad vertices and four bad pairs. Then Phase 3 takes only one step and produces Figure 7, with the 2-circuit processed and all interesting edges restored.

Finally, there are four remaining circuits in $\mathcal{C}$, each consisting of a 4-cycle that can be processed using a single switch. Performing these four switches completes the construction of the canonical path from G to G' with respect to the pairing ψ. (No bad pairs are created during these four final switches.)

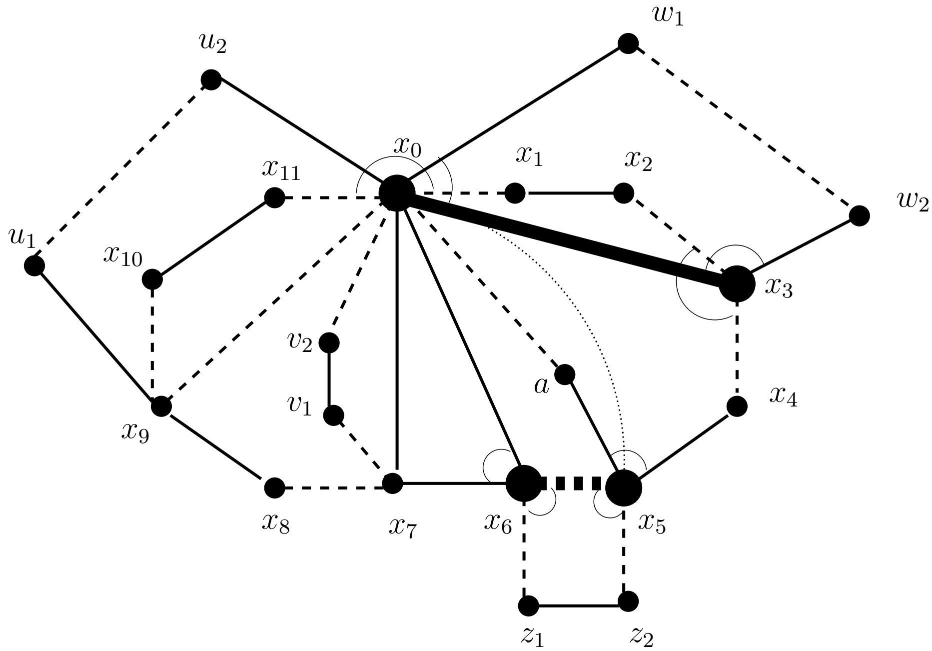

Figure 3: After Phase 1, with 2 interesting edges, 4 bad vertices, 8 bad pairs.

References

[1] C. Cooper, M.E. Dyer and C. Greenhill, Sampling regular graphs and a peer-to-peer network, *Combinatorics, Probability and Computing* **16** (2007), 557–593.

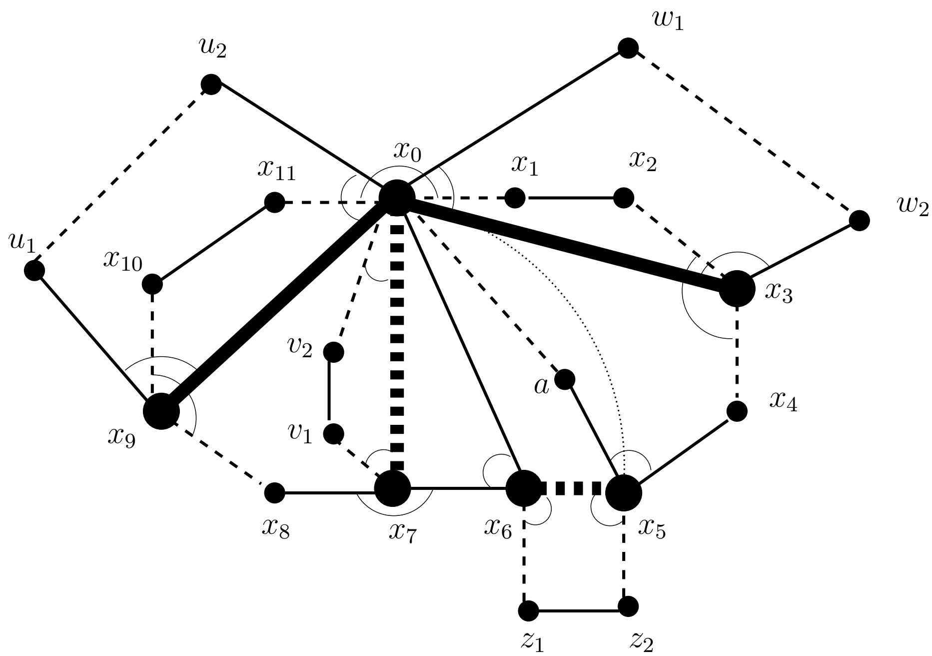

Figure 4: After one step of Phase 2, with 4 interesting edges, 6 bad vertices, 14 bad pairs.

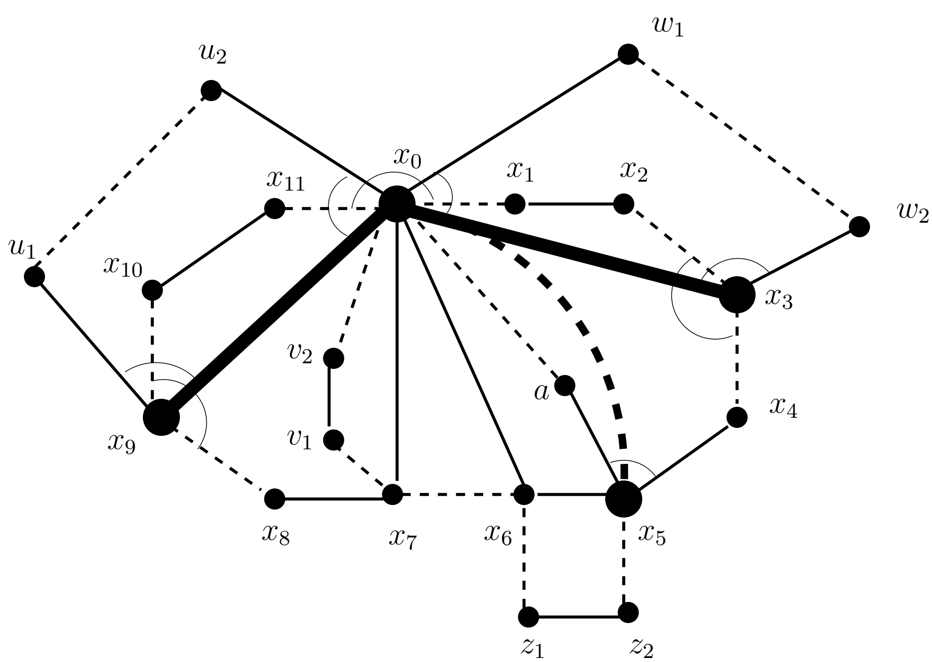

Figure 5: After two steps of Phase 2, with 3 interesting edges, 4 bad vertices, 8 bad pairs.

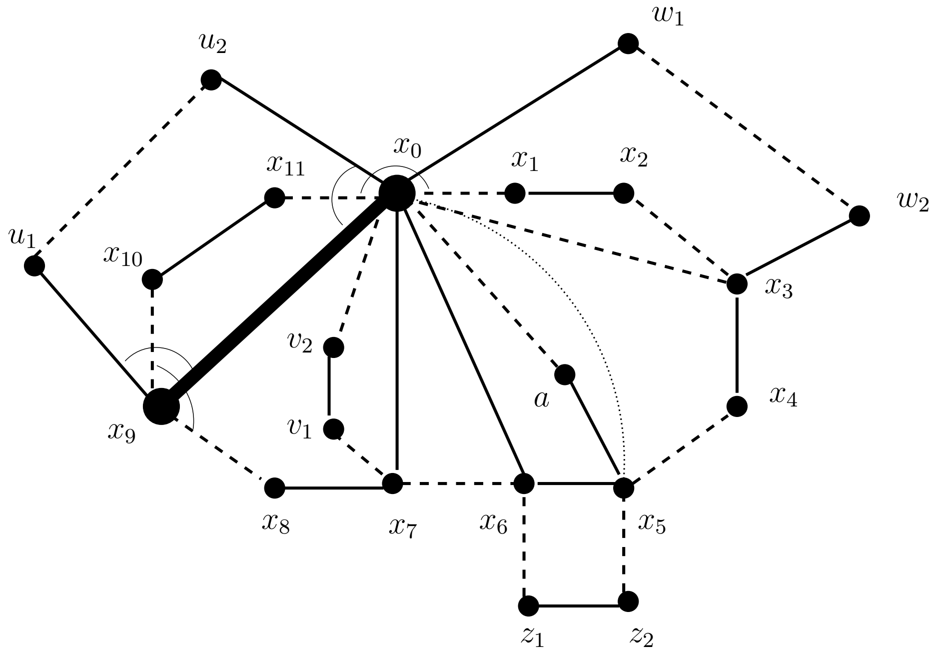

Figure 6: After Phase 2, with 1 interesting edge, 2 bad vertices, 4 bad pairs.

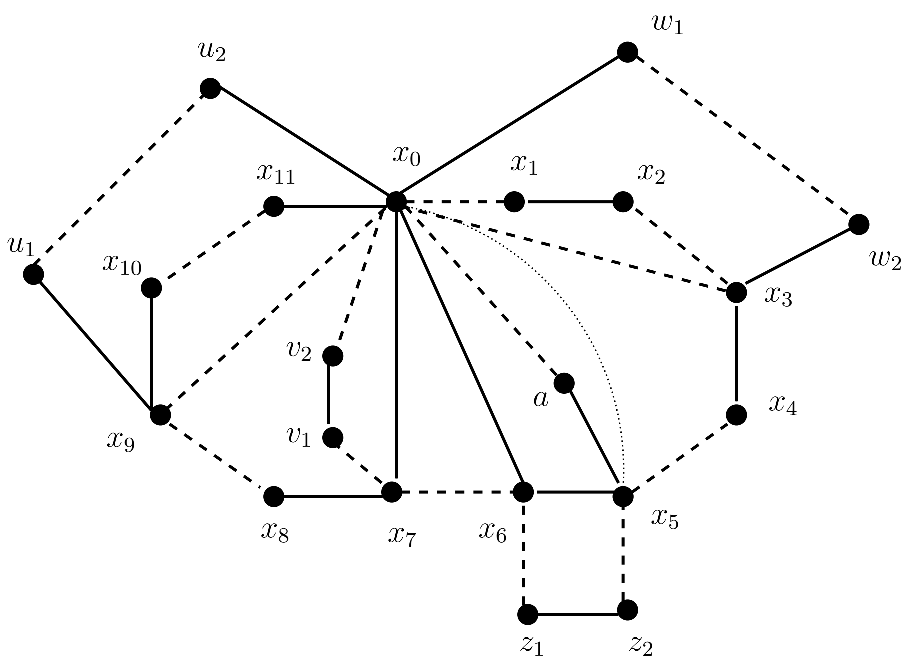

Figure 7: After Phase 3: processing of the 2-circuit is complete.